10 WAYS TO IMPROVE AD SECURITY, QUICKLY.

ABSTRACT

Few things in security are either quick or simple because every environment is different. With so many variables in play across diverse environments, here are a list of widely applicable "easy" wins for securing Active Directory.

Jim Sykora – Identity Security Consultant – Trimarc Security

Trimarc Webcast – Ten Ways to Improve AD Security Quickly: The Blog Post

Introduction

In June of 2022, Trimarc hosted a webinar titled "Top 10 Ways to Improve Active Directory Security Quickly" presented by Sean Metcalf (@PyroTek3), Tyler Robinson (@tyler_robinson), and Darryl Baker (@DFIRdeferred).

The webcast covered several Active Directory attacks and configurations from Blue, Red, and Purple team perspectives to help illustrate why Trimarc makes some of our recommendations in our Active Directory Security Assessment (ADSA) and Enterprise Security Posture (ESP) services. In the webinar, our team demonstrated some of the most valuable recommendations that can be implemented in the least amount of time. From that webinar, we've created the following white paper on improving Active Directory (AD) security quickly.

A slide deck and a recording of the webcast can be found here:
https://www.hub.trimarcsecurity.com/post/webcast-top-10-ways-to-improve-active-directory-security-quickly

Trimarc Recommendations

Few things in security are either quick or simple. Every environment is different, which makes it challenging for Trimarc to make blanket statements of what will work in any environment. Honestly, it should make it hard for any vendor to make blanket statements, but that's another white paper altogether. This is also one of the reasons why Microsoft has a tough time enforcing more secure defaults and configurations in order to maintain backwards or legacy compatibility.

With so many variables in play across different environments, here are a list of widely applicable "easy" wins for securing Active Directory. This article is going to be laying this out in the format of What, Why, When, and How. The Who is going to be your organization.

Passwords: Length = Strength

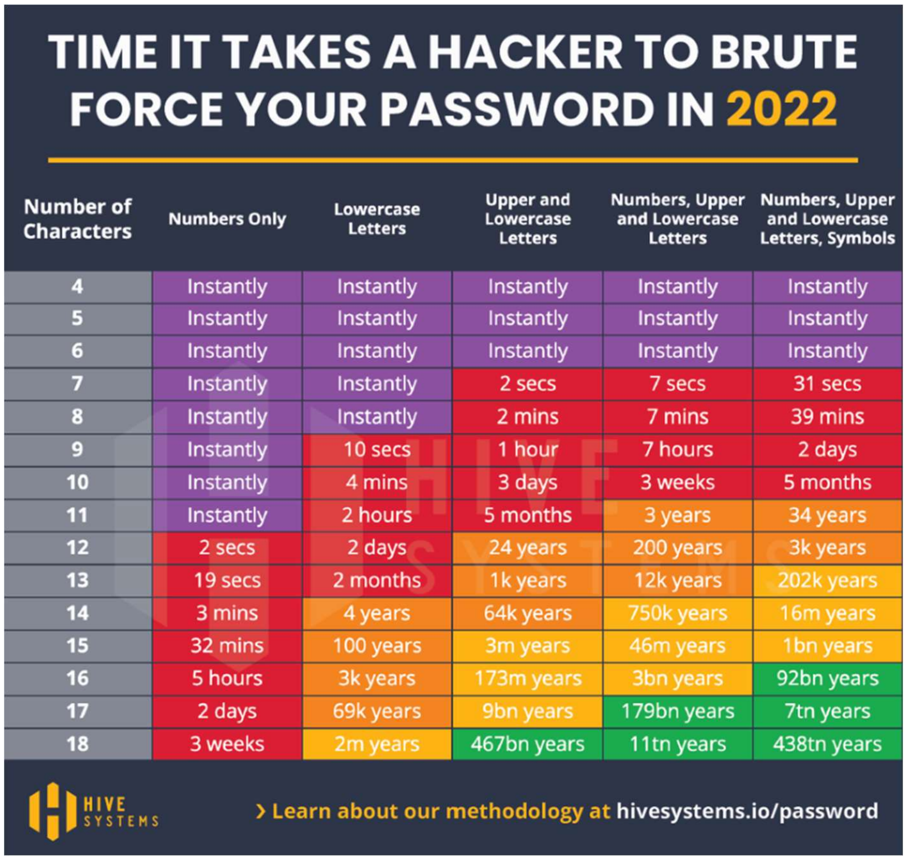
> *Work to move the domain password length up to 12-15 characters. Leverage Fine-Grained Password Policies for admin & service accounts in the near-term.*

What: Migrate Domain Password Policy to 12+ Characters

Trimarc recommends strong passwords with annual rotation. The Domain Password Policy should be set to 12 characters or more, but preferably 15+ characters. Use a password filter like Azure AD Password Protection, if possible, to help prevent known bad passwords like "Password123!", "Summer2022!", or "CompanyName!".

Why:

When it comes to passwords, strength is length. More specifically, strength is in the entropy. There are plenty of "industry best-practices" that auditors and compliance standards tout which state that an 8-character minimum password is good enough. The reality is those standards just haven't kept up with the times. Studies show that if an 8-character minimum password exists, then most users will select an 8-character password. With today's GPU-based cracking technology, an 8-character password, even with all the complexity, can be cracked in under an hour, whereas a 14-character password containing only lowercase letters may take years to crack.



TIME IT TAKES A HACKER TO BRUTE FORCE YOUR PASSWORD IN 2022

Number of Characters	Numbers Only	Lowercase Letters	Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters	Numbers, Upper and Lowercase Letters, Symbols
4	Instantly	Instantly	Instantly	Instantly	Instantly
5	Instantly	Instantly	Instantly	Instantly	Instantly
6	Instantly	Instantly	Instantly	Instantly	Instantly
7	Instantly	Instantly	2 secs	7 secs	31 secs
8	Instantly	Instantly	2 mins	7 mins	39 mins
9	Instantly	10 secs	1 hour	7 hours	2 days
10	Instantly	4 mins	3 days	3 weeks	5 months
11	Instantly	2 hours	5 months	3 years	34 years
12	2 secs	2 days	24 years	200 years	3k years
13	19 secs	2 months	1k years	12k years	202k years
14	3 mins	4 years	64k years	750k years	16m years
15	32 mins	100 years	3m years	46m years	1bn years
16	5 hours	3k years	173m years	3bn years	92bn years
17	2 days	69k years	9bn years	179bn years	7tn years
18	3 weeks	2m years	467bn years	11tn years	438tn years

HIVE SYSTEMS > Learn about our methodology at hivesystems.io/password

Figure 1 2022 Password Brute Force Time (credit Hive Systems)

There's an additional reason for going with 15+ character passwords in Microsoft environments. Due to older defaults for backwards compatibility with older operating systems, some AD environments may still be computing and storing LM hashes, which are less secure than NT hashes. LM hashes can be cracked via rainbow tables in moments. When a password of more than 14 characters is created, the LM hash is not stored. Another way to prevent storing LM hashes is via Group Policy.

Complex, randomly generated credentials are challenging for people to create and remember. So, people end up either creating easy passwords, writing them down, or forgetting them. Easy passwords are easily guessed, password sprayed, or brute forced. Written down passwords might be OK if they're stored in a safe place like the same wallet where you store important bits of paper called money or locked up in a drawer or safe. But when they're taped to monitors or under keyboards those written down passwords can violate the non-repudiation function of credentials by allowing coworkers (or worse) to log in as you. And forgotten passwords create help desk issues and wasted time for the users that are not able to log in.

Those complex, randomly generated credentials do have their place: in a password vault where they can be automated. Password vaults provide a great alternative to password reuse. Password reuse is when the same password is set across multiple services. When a reused password gets breached one place, the time to compromise it elsewhere is a number approaching zero. A password vault doesn't have issues generating, remembering, or auto-filling unique, strong, complex passwords across multiple systems. Just be sure to select a long passphrase for the vault access credential and set up MFA.

Start thinking in terms of long passphrases, like the classic XKCD example "Correct Horse Battery Staple", and less in terms of passwords for human accounts. Passphrases are generally easier for humans to remember than random passwords and have more entropy than short, random passwords.

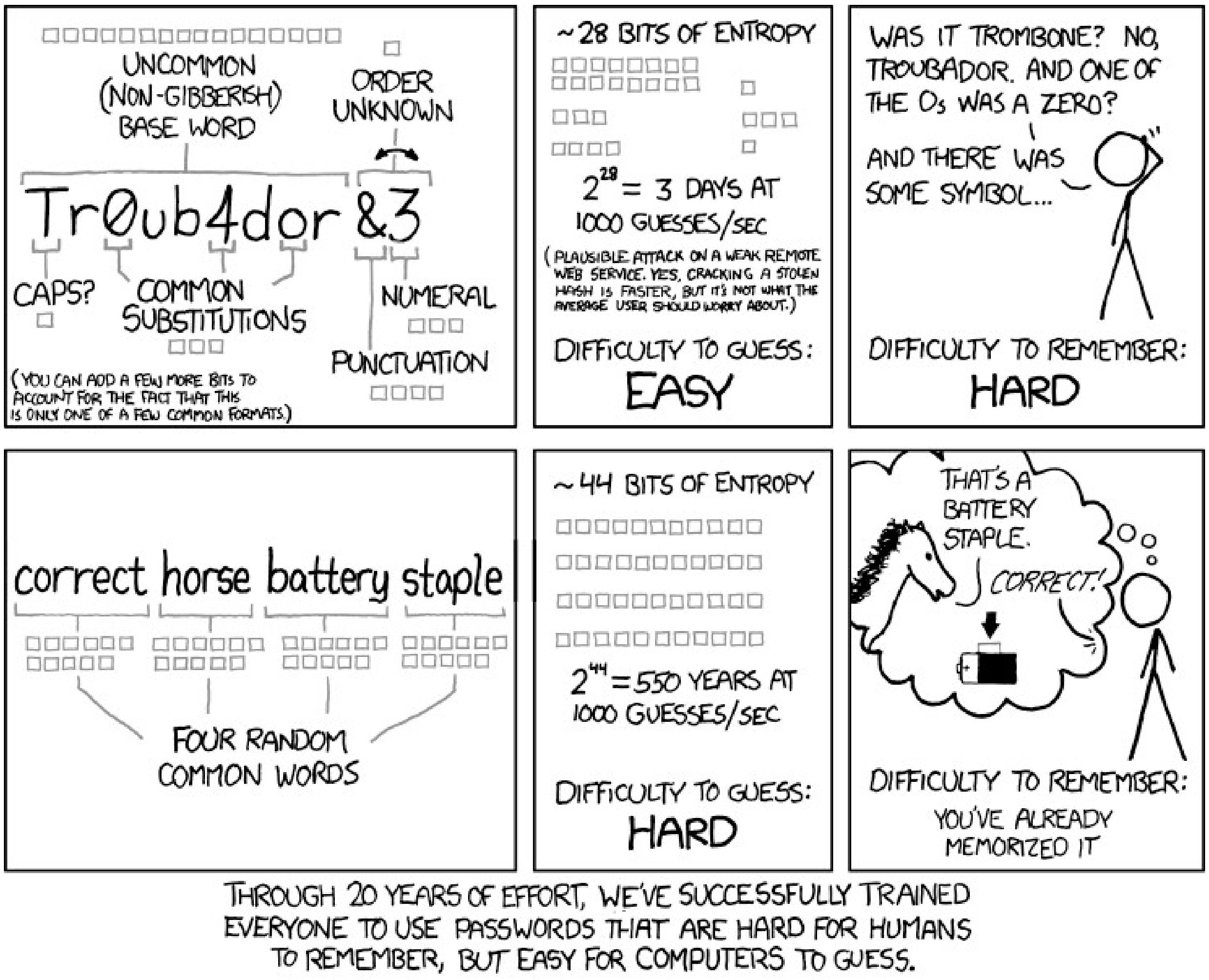

UNCOMMON (NON-GIBBERISH) BASE WORD
ORDER UNKNOWN
Tr0ub4dor &3
CAPS?
COMMON SUBSTITUTIONS
NUMERAL
PUNCTUATION
(YOU CAN ADD A FEW MORE BITS TO ACCOUNT FOR THE FACT THAT THIS IS ONLY ONE OF A FEW COMMON FORMATS.)
~28 BITS OF ENTROPY
2^28 = 3 DAYS AT 1000 GUESSES/SEC
(PLAUSIBLE ATTACK ON A WEAK REMOTE WEB SERVICE. YES, CRACKING A STOLEN HASH IS FASTER, BUT IT'S NOT WHAT THE AVERAGE USER SHOULD WORRY ABOUT.)
DIFFICULTY TO GUESS: EASY
WAS IT TROMBONE? NO, TROUBADOR. AND ONE OF THE Os WAS A ZERO?
AND THERE WAS SOME SYMBOL...
DIFFICULTY TO REMEMBER: HARD
correct horse battery staple
FOUR RANDOM COMMON WORDS
~44 BITS OF ENTROPY
2^44 = 550 YEARS AT 1000 GUESSES/SEC
DIFFICULTY TO GUESS: HARD
THAT'S A BATTERY STAPLE.
CORRECT!
DIFFICULTY TO REMEMBER: YOU'VE ALREADY MEMORIZED IT
THROUGH 20 YEARS OF EFFORT, WE'VE SUCCESSFULLY TRAINED EVERYONE TO USE PASSWORDS THAT ARE HARD FOR HUMANS TO REMEMBER, BUT EASY FOR COMPUTERS TO GUESS.

Figure 2 Password Strength (Credit xkcd: https://xkcd.com/936/)

With greater password strength, there's less of a need for frequent password rotation such as the old-school 90 days standard that PCI DSS prescribes. The primary reason for regularly rotating credentials is because they might be compromised. But the 90-day rotation doesn't add up if an 8-character password can be cracked in under 90 minutes. If stronger passwords are enforced and users educated on the dangers of password re-use, it's far less likely that credentials will be compromised. Some auditors may struggle with this but try walking them through NIST 800-63b. Etch your organization's password policy not just in GPO, but in a policy document approved by Senior Management and/or the Board.

When:

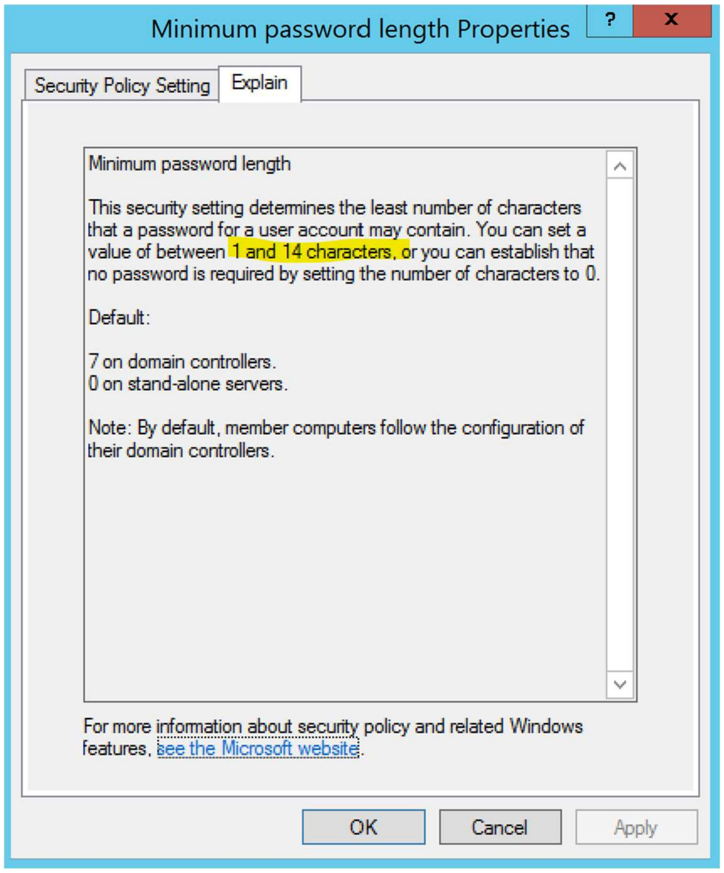
Disclaimer: The Passwords section is out of order because it made more sense to explain all the "Why" around longer credentials first, so implement your Domain Password Policy after implementing longer Fine-Grained Password Policies on AD Admins and non-human service accounts.

Do some org-wide end-user training before changing the policy on:

- How longer passphrases are easier to remember while also more secure.
- The dangers of password re-use.
- How longer passphrases allow the organization to set a policy where password changes won't happen as frequently.

How:

There are some oddities around how password policies work in modern Windows OSes. If the goal is to set a 12- or 14-character minimum password length, this can all be done via GPO. But if you want to go for the gusto and jump to a 15+ character minimum password length you may notice this:



Minimum password length

This security setting determines the least number of characters that a password for a user account may contain. You can set a value of between 1 and 14 characters, or you can establish that no password is required by setting the number of characters to 0.

Default:

7 on domain controllers.
0 on stand-alone servers.

Note: By default, member computers follow the configuration of their domain controllers.

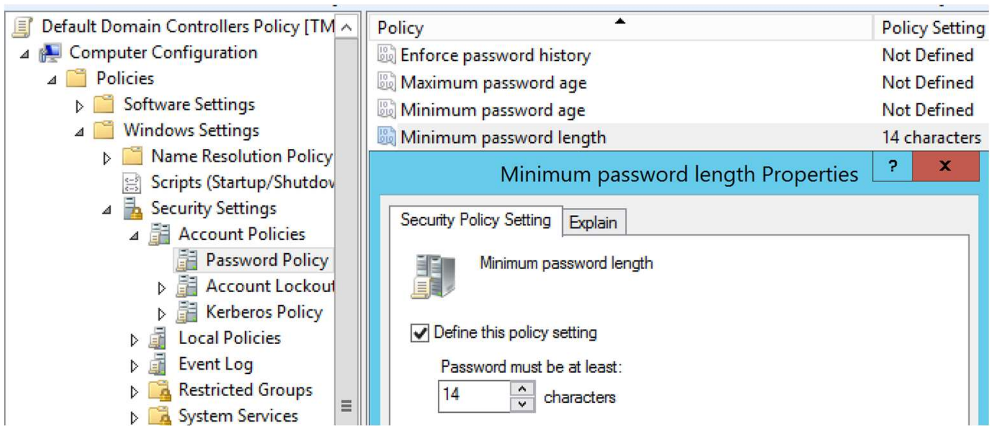
For more information about security policy and related Windows features, see the Microsoft website.

Figure 3 Minimum password length properties

Configuring a minimum password length of more than 14 characters requires Fine-Grained Password Policies (FGPP), unless all DCs are running Windows Server v2004 or later (including Windows Server 2022). Fine-Grained Password Policies require at least Windows Server 2008 AD DS, but to configure a FGPP via GUI requires at least Windows Server 2012.

For password policies up to 14 characters in length:

Edit the Default Domain Controllers Policy in Group Policy Management, navigate to Computer Configuration->Policies->Windows Settings->Security Settings->Account Policies->Password Policy->Minimum password length. Define this policy setting and set it to a number between 12 and 14 characters.



Policy	Policy Setting
Enforce password history	Not Defined
Maximum password age	Not Defined
Minimum password age	Not Defined
Minimum password length	14 characters

Minimum password length

☑ Define this policy setting

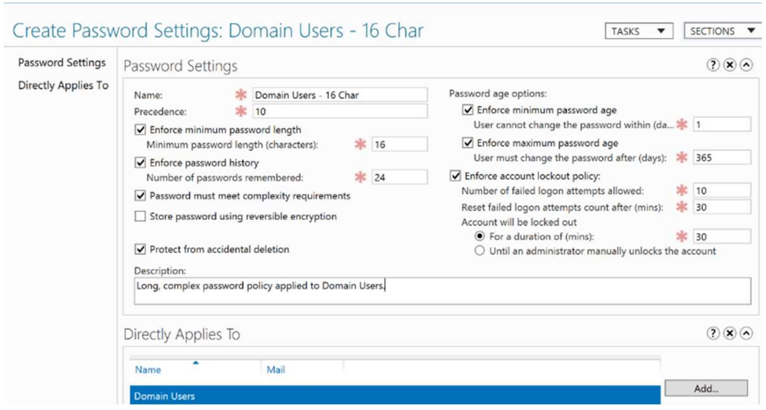
Password must be at least: 14 characters

Figure 4 Group Policy: Password Length

For password policies more than 14 characters in length:

This will require Fine-Grained Password Policies configured in Active Directory Administration Center.

1. Open AD Administrative Center with Domain Admin privileges.
2. Navigate to <Domain>->System->Password Settings
3. Right-click then New->Password Settings



Password Settings

Name: Domain Users - 16 Char
Precedence: 10
☑ Enforce minimum password length
Minimum password length (characters): 16
☑ Enforce password history
Number of passwords remembered: 24
☑ Password must meet complexity requirements
☐ Store password using reversible encryption
☑ Protect from accidental deletion

Password age options:
☑ Enforce minimum password age
User cannot change the password within (da...: 1
☑ Enforce maximum password age
User must change the password after (days): 365
☑ Enforce account lockout policy:
Number of failed logon attempts allowed: 10
Reset failed logon attempts count after (mins): 30
Account will be locked out
● For a duration of (mins): 30
○ Until an administrator manually unlocks the account

Description:
Long, complex password policy applied to Domain Users

Directly Applies To

Name	Mail
Domain Users	

Figure 5 Domain Users - 16 Char FGPP Example

In the above graphic we've created a Password Setting named "Domain Users – 16 Char" with a Precedence of 10. A higher Precedence is important because we want the lower Precedence policies that are applied to AD Admins and service accounts to take effect. We've set a 16-character minimum password length, enforced password history, and complexity. In this example we also set Password age options and account lockout policy.

Something to consider with the maximum password age set here (or what exists in your Default Domain Controllers Policy) is that the new password requirements won't take effect until each user changes their password. It may be prudent to start out with an enforced maximum password age of 60-90 days to ensure the new 16 character minimum applies to all your users. Then set a reminder and use AD PowerShell Module to look for accounts that haven't changed passwords yet since the new FGPP was applied and resolve those accounts before changing to a longer maximum password age.

What: Configure Fine-Grained Password Policies

Fine-Grained Password Policies allow for implementing better password policies in phases by applying different policies to distinct groups.

The two most important "groups" to start with are AD Admin accounts and non-human service accounts.

Why:

AD Admins are highly privileged accounts and need to be protected according to that privilege. This makes them a great steppingstone for going with longer credentials everywhere.

For service accounts we're looking to make guessing, cracking, and Kerberoasting infeasible. Since service accounts aren't human accounts, nobody needs to type them in regularly. Except when they're rotated annually. This makes 30 character randomly generated passwords a good fit. Store them in a **secure** password vault or printed out and in a literal vault. When and wherever possible, use Group Managed Service Accounts instead of manually managed service accounts. This DFIR Report article shows how a threat actor utilized Kerberoasting a service account with Domain Admin privileges during an attack: https://thedfirreport.com/2022/08/08/bumblebee-roasts-its-way-to-domain-admin/

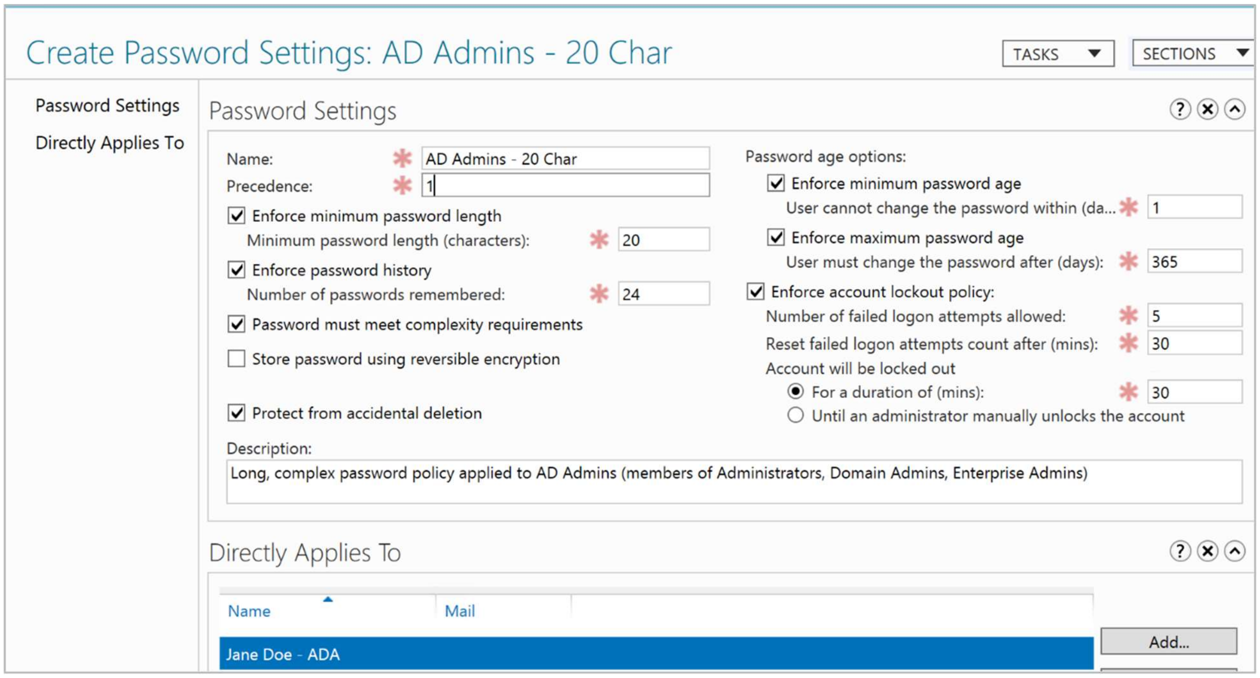
When:

AD Admin accounts FGPP should be done right away, before you force AD Admin password rotation for the first time, before you tackle service accounts, and certainly before you set a domain-wide password policy.

How:

1. Create 2 new security groups with a Global scope. Groups that control application of Fine-Grained Password Policies are privileged and should be created in top-level Tier 0 OUs that only AD Admins have rights to. Follow your org's group naming scheme, but here are examples:
    a. FGPP_ADAdmin - Members: Domain Admins, Administrator, and individually any other AD Admin accounts that are members of or delegated highly privileged Tier 0 access to AD.
    b. FGPP_ServiceAccounts - Members: Add service accounts individually starting with a test account and expanding until all service accounts are added.
2. Open Active Directory Administrative Center with Domain Admin privileges
3. Create a new Password Setting by navigating to System->Password Settings, right-click, New|Password Settings

a. The most important settings here are the Precedence & Enforce minimum password length. The Precedence applies settings in order when there are more than one. Whichever Password Setting has the lowest Precedence will be applied first. This is important in instances where a user might have more than one policy applied. Any setting that is not explicitly configured here will fall back to the Password Policy configured in the Default Domain Controllers Policy GPO.


Create Password Settings: AD Admins - 20 Char

Password Settings

Name: AD Admins - 20 Char
Precedence: 1
[✓] Enforce minimum password length
Minimum password length (characters): 20
[✓] Enforce password history
Number of passwords remembered: 24
[✓] Password must meet complexity requirements
[ ] Store password using reversible encryption
[✓] Protect from accidental deletion

Password age options:
[✓] Enforce minimum password age
User cannot change the password within (da... 1
[✓] Enforce maximum password age
User must change the password after (days): 365
[✓] Enforce account lockout policy:
Number of failed logon attempts allowed: 5
Reset failed logon attempts count after (mins): 30
Account will be locked out
● For a duration of (mins): 30
○ Until an administrator manually unlocks the account

Description:
Long, complex password policy applied to AD Admins (members of Administrators, Domain Admins, Enterprise Admins)

Directly Applies To

Name	Mail
Jane Doe - ADA	

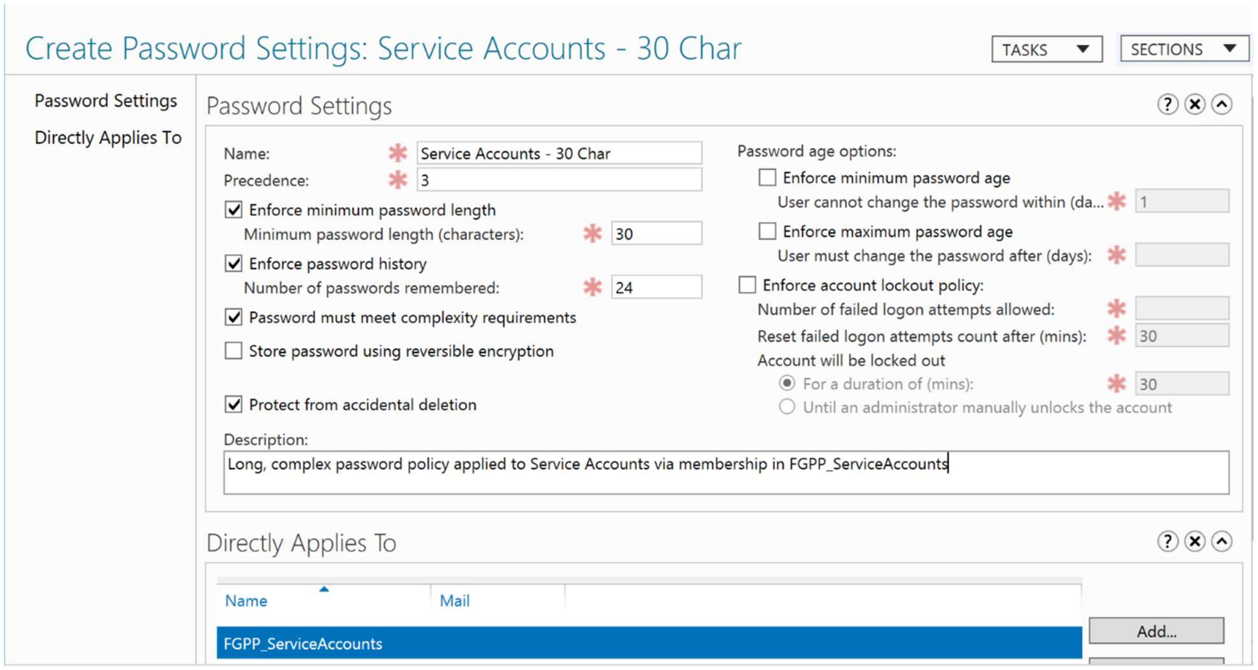
Figure 6 AD Admins - 20 Char FGPP Example

b. Here we've created a new Password Policy named "AD Admins – 20 Char" with a Precedence of 1 (because we want this policy to apply first). It enforces a minimum password length of 20 characters, password history of 24, complexity requirements, minimum password age of 1 day, and maximum password age of 365* days. We've also configured an account lockout policy here of 5 failed attempts where the counter resets after 30 minutes and if the account is locked out it lasts for 30 minutes. You may want to weigh the benefits of all AD Admin accounts having an automatic lockout policy with the risk of a targeted denial-of-service attack on those accounts.

c. We recommend you start the AD Admins policy with a maximum password age of 30 days and then set a reminder to monitor and use AD PowerShell Module to look for accounts that haven't changed passwords yet since the new FGPP was applied. Resolve any password age issues on those accounts before changing to a longer maximum password age. The new password requirements won't take effect until a user changes their password and immediately setting a long maximum password age won't help with that.

d. We've only applied the new AD Admins – 20 Char password policy to 1 individual user account at this point. First do some testing with one account to make sure there are no legacy compatibility issues in your environment. Rotate the password on your test account so it conforms to the new password policy, wait briefly for the new credential to replicate, and perform some administrative functions with that account before adding your version of the "FGPP_ADAdmins" group to the AD Admin Password Setting.

4. Create another new Password Setting for service accounts.



Password Settings

Name: Service Accounts - 30 Char
Precedence: 3
☑ Enforce minimum password length
Minimum password length (characters): 30
☑ Enforce password history
Number of passwords remembered: 24
☑ Password must meet complexity requirements
☐ Store password using reversible encryption
☑ Protect from accidental deletion

Password age options:
☐ Enforce minimum password age
User cannot change the password within (da...: 1
☐ Enforce maximum password age
User must change the password after (days):
☐ Enforce account lockout policy:
Number of failed logon attempts allowed:
Reset failed logon attempts count after (mins): 30
Account will be locked out
● For a duration of (mins): 30
○ Until an administrator manually unlocks the account

Description: Long, complex password policy applied to Service Accounts via membership in FGPP_ServiceAccounts

Directly Applies To

Name	Mail
FGPP_ServiceAccounts	

Add...

Figure 7 Service Accounts – 30 Char FGPP Example

a. Here we're applying a 30-character minimum password length and have set a Precedence of 3, meaning that if an account is a member of both FGPP_ADAdmins and FGPP_ServiceAccounts the "AD Admins – 20 Char" policy would apply instead of this one. We've left off the Password age options as although it's important to rotate service account credentials annually, this should be done without expiring the credentials for availability reasons.
b. Add one service account to your version of the FGPP_ServiceAccounts group and manually rotate its credentials so the new policy takes effect. Set the new credentials in the service, task, or wherever else it's being used and make sure everything is working before doing the same with additional service accounts.
c. Remember, the new password length requirements will not take effect until these accounts change their password with the new policy applied. For non-human service accounts, you will need to go through each account and carefully rotate the credentials. This is so you don't have any accounts with old, weak passwords.

5. If you run into issues with enforcing credentials longer than 14 characters, review the `MinimumPasswordLengthAudit` auditing measures made available in Windows Server 2016+ via this article: https://support.microsoft.com/en-us/topic/minimum-password-length-auditing-and-enforcement-on-certain-versions-of-windows-5ef7fecf-3325-f56b-cc10-4fd565aacc59

More Resources:

- https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/get-started/adac/introduction-to-active-directory-administrative-center-enhancements--level-100-#fine_grained_pswd_policy_mgmt
- https://en.wikipedia.org/wiki/Password_strength
- https://www.hivesystems.io/blog/are-your-passwords-in-the-green
- https://docs.microsoft.com/en-us/archive/blogs/secguide/security-baseline-final-for-windows-10-v1903-and-windows-server-v1903
- https://pages.nist.gov/800-63-3/sp800-63b.html
- https://xkcd.com/936/
- https://docs.microsoft.com/en-us/windows-server/security/kerberos/passwords-technical-overview
- https://docs.microsoft.com/en-US/troubleshoot/windows-server/windows-security/prevent-windows-store-lm-hash-password
- https://www.blackhillsinfosec.com/webcast-passwords-you-are-the-weakest-link/
- https://www.blackhillsinfosec.com/increase-minimum-character-password-length-15-policies-active-directory/
- https://adsecurity.org/?p=2293
- https://support.microsoft.com/en-us/topic/minimum-password-length-auditing-and-enforcement-on-certain-versions-of-windows-5ef7fecf-3325-f56b-cc10-4fd565aacc59

AD Admin Hygiene

Review AD Admin group membership regularly, enforce annual password changes, and remove inactive AD Admin accounts.

What: Review AD Admin Group Membership Regularly

If you don't have a straightforward way to identify your AD admins, you're already behind the 8 ball. Ensure there is a documented process for regularly reviewing the membership of Domain Admins, Administrators, Enterprise Admins (aka the AD Admin groups). Every member should map to a verifiable human being. Regular review of privileged group membership needs to be a regularly scheduled process.

Why:

Trimarc ADSA assessments regularly find accounts that don't belong in the most privileged AD Admin groups. These standard user accounts, computers, service accounts, and nested groups mixed into the most privileged access groups in AD allow an attacker to go straight from compromised user to compromised AD.

When: Start today and then annually at a minimum.

How:

There are several ways to do this:

- Manually review group membership with Active Directory Users and Computers (ADUC) 🦆
- Leverage AD PowerShell module:

  ```
  Get-ADGroupMember 'Administrators' -Recursive
  ```

- Utilize the free Trimarc PowerShell script Invoke-TrimarcADChecks.ps1

What: Enforce Annual Password Changes

There are a few different standards for password age and rotation. PCI requires users to change their passwords every 90 days. This leads to people using weak passwords like "Summer2022!" or "P@ssword73". NIST 800-63b famously provides guidance for passwords that never expire unless there is a known indication of compromise. That's only if your policy follows **all** the other 800-63b requirements which, unfortunately, few organizations are.

Trimarc recommends a middle ground of strong passwords with annual rotation. When it comes to passwords, the strength is in the length. The Domain Password Policy should be set to 12 characters or more, but preferably 15+ characters with annual rotation. Use a password filter like Azure AD Password Protection if you can.

Why:

Trimarc regularly finds accounts, including AD Admins, that haven't had a password changed in years. Stale passwords may follow older, less secure password policies or may already be compromised. You shouldn't have Service Accounts that are members of AD Admins, but Service Accounts with Service Principal Names (SPNs) are susceptible to Kerberoasting if their passwords are stale and weak (short).

When:

Configure FGPP for AD Admins first. Re-check for stale passwords and `PasswordNeverExpires` at least annually.

How:

1. Set a reasonable password policy that will take effect the next time users change their password.
2. Check for accounts configured with Password Never Expires leveraging AD PowerShell module:

   ```
   Get-ADUser -filter * -properties Name, PasswordNeverExpires | Where-
   Object {$_.PasswordNeverExpires -eq "true" } |  Select-Object
   DistinguishedName,Name,EnabledPasswordLastSet,Description,Created,Us
   erPrincipalName | Where-Object {($_.LastLogonDate -le $InactiveDate)
   -AND ($_.PasswordLastSet -le $InactiveDate)}
   ```

3. Resolve issues with flagged accounts. Pay special attention to service accounts, although these should be rotated at least annually as well.
4. Check for accounts with a password more than one year old using AD PowerShell module:

   ```
   Get-ADUser -filter * -property
   SAMAccountName,DisplayName,LastLogonDate,PasswordLastSet,Description
   ,Created,UserPrincipalName | Where-Object {($_.PasswordLastSet -le
   (Get-Date).AddDays(-365))}
   ```

What: Remove Inactive AD Admin Accounts

Unused or underutilized accounts in highly privileged groups, outside of any break-glass emergency accounts like the default Administrator account, should have their AD Admin privileges removed.

Why:

Trimarc regularly discovers highly privileged accounts that haven't been utilized for years. These unused or maybe forgotten accounts are prime targets for attackers. Few would notice that an attacker had changed the credentials on their unused AD Admin account.

When:

Check for inactive accounts now and recheck quarterly.

How:

Use the AD PowerShell module to find inactive AD Admins that haven't logged on in the past 30 days and haven't changed their password in the past year. Then determine if the resulting accounts can be safely removed from AD Admin group membership.

```
Get-ADGroupMember 'Administrators' -Recursive | Get-ADUser -property
SAMAccountName,DisplayName,LastLogonDate,PasswordLastSet,Description,Created,
UserPrincipalName | Where-Object {($_.LastLogonDate -le (Get-Date).AddDays(-
30)) -AND ($_.PasswordLastSet -le (Get-Date).AddDays(-365))}
```

More Resources:

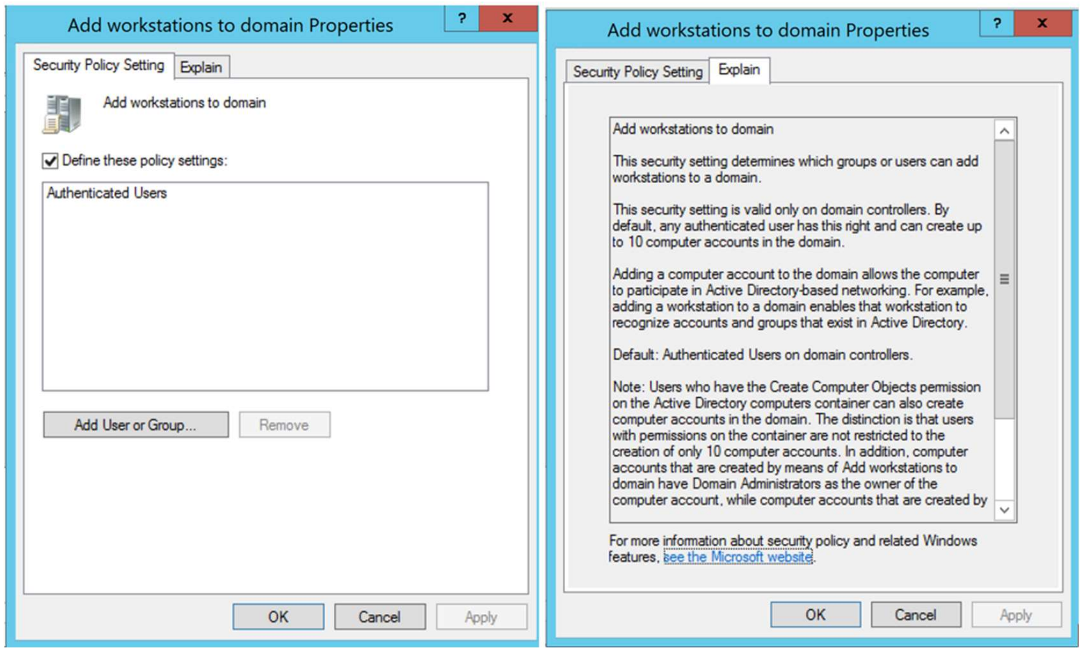
- https://adsecurity.org/?p=3700
- https://www.hub.trimarcsecurity.com/post/implementing-controls-in-active-directory-protecting-against-privileged-credential-sprawl
- https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/implementing-least-privilege-administrative-models
- https://www.hub.trimarcsecurity.com/post/there-s-something-about-service-accounts
- https://pages.nist.gov/800-63-3/sp800-63b.html

Restrict Domain Join

Restrict accounts that are allowed to add workstations to the domain via Machine Account Quota and/or the SeMachineAccountPrivilege.

What: By Default, Authenticated Users Can Add PCs to Domain

The default "Default Domain Controllers Policy" GPO assigns members of Authenticated Users the `SeMachineAccountPrivilege` right to add up to `ms-DS-MachineAccountQuota` computer objects to an Active Directory Domain. The default `ms-DS-MachineAccountQuota` is 10.


Add workstations to domain Properties

Security Policy Setting | Explain

Add workstations to domain

[✓] Define these policy settings:

Authenticated Users

Add workstations to domain

This security setting determines which groups or users can add workstations to a domain.

This security setting is valid only on domain controllers. By default, any authenticated user has this right and can create up to 10 computer accounts in the domain.

Adding a computer account to the domain allows the computer to participate in Active Directory-based networking. For example, adding a workstation to a domain enables that workstation to recognize accounts and groups that exist in Active Directory.

Default: Authenticated Users on domain controllers.

Note: Users who have the Create Computer Objects permission on the Active Directory computers container can also create computer accounts in the domain. The distinction is that users with permissions on the container are not restricted to the creation of only 10 computer accounts. In addition, computer accounts that are created by means of Add workstations to domain have Domain Administrators as the owner of the computer account, while computer accounts that are created by

For more information about security policy and related Windows features, see the Microsoft website.

Figure 8 Add workstations to domain Properties

When a non-privileged, non-delegated user creates a computer account utilizing `SeMachineAccountPrivilege` the creating account's SID is stored in the computer account's *ms-DS-CreatorSID*.

Why:

Part of the text that's left off in the screenshot above is "computer accounts that are created by means of permissions on the computers container have the creator as the owner of the computer account."

This means that when a regular user creates a computer account in AD, they are set as the owner on the object. This assigns several rights on the computer account to the user, including the ability to modify computer attributes and read extended attributes.

When an attacker creates their own computer account, they can choose the password of the computer account and with ownership can modify the computer account's own *msDS-*`AllowedToActOnBehalfOfOtherIdentity` attribute and other attributes. This allows for the use of Resource-Based Constrained Delegation (RBCD) and Services for User to Self (S4U2self) to impersonate an arbitrary user to that computer. This is one of the techniques KrbRelayUp utilizes in the attack chain ending with local privilege escalation to SYSTEM. An attacker can also use computer account credentials when executing scripts or code in the environment, which can go undetected by some SIEM configurations.

When:

Soon

How:

1. Review the status of the `SeMachineAccountPrivilege` using Group Policy Management console:

   a. Default Domain Controllers Policy->Computer Configuration->Policies->Windows Settings->Security Settings->Local Policies->User Rights Assignment->Add workstations to domain
   b. Note: If your organization followed best practices and didn't modify the Default Domain Controllers Policy directly, perform some Group Policy Modeling on the Domain Controllers container to determine which policy performs User Rights Assignments there.
2. Use AD PowerShell module to check the current `ms-DS-MachineAccountQuota`:

```
Get-ADObject ((Get-ADDomain).distinguishedname) -Properties ms-DS-
MachineAccountQuota | select ms-DS-MachineAccountQuota
```

3. Consider how IT adds computers to the domain. Are new workstations added by IT staff? Are they using specific admin accounts with correctly delegated permissions for this purpose?
4. Remove the capability for regular users to add workstations to the domain:
   a. Remove "Authenticated Users" from the Add workstations to domain user rights assignment in the Default Domain Controllers Policy
   b. Set the `ms-DS-MachineAccountQuota` to 0 with AD PowerShell:

   ```
   Set-ADDomain (Get-ADDomain).distinguishedname -Replace @{"ms-DS-
   MachineAccountQuota"="0"}
   ```

5. Use AD PowerShell module to find computer accounts added by regular users:

```
Get-ADObject -LDAPFilter "(mS-DS-CreatorSID=*)"  -Properties * |
Format-Table ObjectClass, Name, DistinguishedName, mS-DS-CreatorSID -
AutoSize
```

   a. Map a SID from `ms-DS-CreatorSID` to an account:

   ```
   $sid = 'S-1-5-21-2914015308-3349062846-2830758205-101609'#example

   Get-ADObject –IncludeDeletedObjects -Filter "objectSid -eq
   '$sid'" | Select-Object name, objectClass
   ```

6. Use AD PowerShell module to find computer accounts that aren't owned by Domain Admins:

```
Get-ADComputer -Filter * -Properties ntSecurityDescriptor | Select-
Object -Property Name, @{Name='ntSecurityDescriptorOwner';
Expression={$_.ntSecurityDescriptor.Owner }}, DistinguishedName | where
{ $_.ntSecurityDescriptorOwner -notlike "*\Domain Admins" }
```

More Resources:

- https://social.technet.microsoft.com/wiki/contents/articles/5446.active-directory-how-to-prevent-authenticated-users-from-joining-workstations-to-a-domain.aspx
- https://www.netspi.com/blog/technical/network-penetration-testing/machineaccountquota-is-useful-sometimes/
- https://shenaniganslabs.io/2019/01/28/Wagging-the-Dog.html
- https://blog.harmj0y.net/redteaming/another-word-on-delegation/
- https://dirkjanm.io/abusing-forgotten-permissions-on-precreated-computer-objects-in-active-directory/

Kerberos Delegation

Review accounts that have Unconstrained Delegation and remove any with no associated Kerberos SPN.

What: Review Accounts with Unconstrained Delegation & Remove any without SPNs

Delegation in Kerberos is a legitimate method for services and accounts to impersonate other accounts. It's a necessary function within Active Directory authentication, but when improperly configured or secured it can be a nasty security situation.

Why:

Unconstrained Delegation is the oldest and least secure form of Kerberos delegation. The biggest flaw in this type of delegation is in its name: it's not constrained. Unconstrained delegation can be used to impersonate any account, including AD Admins or Domain Controllers, if those accounts not properly protected.

When a user authenticates to a system with unconstrained delegation, the system can impersonate that user to any other Kerberos service on the network. If the user account that authenticates to the system's service was a member of Domain Admins, then the system can impersonate the account with Domain Admin rights to the Domain Controller. Attackers know how to coerce and capture authentication from high value targets in order to abuse this impersonation.

When:

Soon. Then review at least annually.

How:

1. Review accounts that have Unconstrained Delegation using AD PowerShell:

   ```
   Get-ADComputer -Filter {(TrustedForDelegation -eq $True) -AND
   (PrimaryGroupID -eq 515)} -Properties
   TrustedForDelegation,TrustedToAuthForDelegation,servicePrincipalName,De
   scription
   ```

   Better yet, use the free Trimarc PowerShell script Invoke-TrimarcADChecks.ps1
2. Further review that query to look for Unconstrained Delegation without an associated SPN and remove the delegation.

Bonus Win: Convert Unconstrained Delegation to Constrained, Utilize the Protected Users group for all AD Admins. Ensure all Service Accounts have long (25+ characters) passwords or use Group Managed Service Accounts (gMSAs)

More Resources:

- Unconstrained Delegation: https://adsecurity.org/?p=1667
- SPN Scanning: https://adsecurity.org/?p=2535

Protect AD Admins from Kerberos Delegation

Configure all AD admin accounts with "Account is Sensitive and cannot be delegated." Then add to the Protected Users group

What: Configure AD Admins with Account is Sensitive and cannot be delegated (Phase 1)

There are two ways to protect AD user accounts from being inadvertently or maliciously delegated for Kerberos authentication. The original method is to set the "Account is sensitive and cannot be delegated" flag on a user account. Enabling this setting on all AD Admin accounts is a quick win for most environments.

Why:

When an account is not specifically protected, it can be delegated (impersonated) by any form of Kerberos delegation. Setting AD Admin accounts as Sensitive is one of the few protections against Unconstrained Delegation abuse. It also protects against abuse of classic “Constrained Delegation” and the newer “Constrained Delegation with Protocol Transition”. However, this flag does not protect against Resource Based Constrained Delegation abuses.

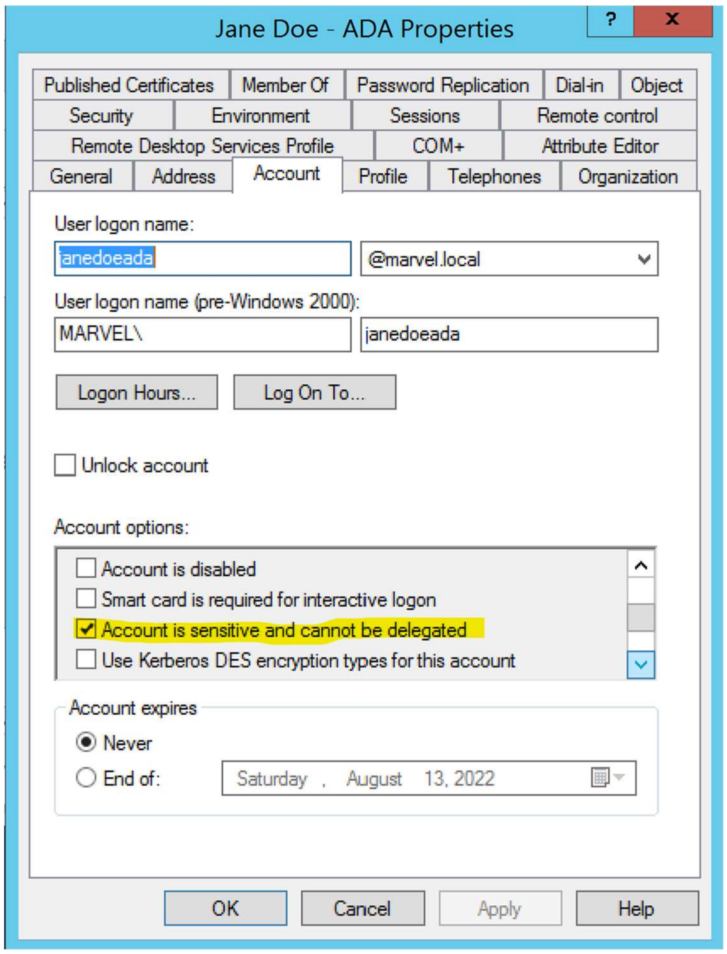
When:

Soon. Review annually to make sure all AD Admin accounts are still flagged sensitive.

How:

Since you’ve done some AD Admin Hygiene in a previous step, you should have very few AD Admins now, right? So, it’s not too bad to use ADUC to manually set this flag on accounts and then check your progress with AD PowerShell module or Invoke-TrimarcADChecks.ps1.

Go through the list of AD Admins that you discovered earlier and open each of them in ADUC. Then go to the Account tab and check “Account is sensitive and cannot be delegated”



User logon name: janedoeada @marvel.local

User logon name (pre-Windows 2000): MARVEL\ janedoeada

☐ Unlock account

Account options:

☐ Account is disabled
☐ Smart card is required for interactive logon
☑ Account is sensitive and cannot be delegated
☐ Use Kerberos DES encryption types for this account

Account expires

● Never
○ End of: Saturday , August 13, 2022

Figure 9 Account is sensitive and cannot be delegated

Warning: Be mindful of any non-human service accounts within AD Admins. There shouldn't be any, but we're only tackling easy improvements in this document. Preventing delegation is likely to break service accounts.

What: Add AD Admins to Protected Users (Phase 2)

Once you've got all your AD Admins flagged as sensitive, start working towards getting them into the Protected Users group, which provides for even more protections on these accounts. Do this in phases with thorough testing.

The protections provided via membership in the Protected Users group is additive to the "Account is sensitive and cannot be delegated" account flag. Do both because there are some scenarios where an account may need to be temporarily removed from Protected Users to perform one specific task. With "Account is sensitive and cannot be delegated" configured, this account will remain protected from delegation attacks for the short period the task is being performed.

Why:

On at least a Windows Server 2012 R2 (or Windows 8.1) with a domain functional level of at least Windows Server 2012 R2 membership in Protected Users grants the following protections:

- Default credential delegation (CredSSP) plaintext creds are not cached
- Windows Digest plaintext credentials are not cached
- NTLM NT One-Way Function (NTOWF) is not cached
- Kerberos Ticket Granting Ticket (TGT) is acquired at logon and can't be re-acquired automatically
- Sign-on offline – cached logon verifier not created
- No NTLM authentication
- No DES or RC4 cipher suites in Kerberos pre-auth
- No delegation by unconstrained or constrained delegation
- No renewal of user TGTs beyond initial 4-hour lifetime
- `TokenLeakDetectDelaySecs` behavior

When:

After all AD Admins are flagged "Account is sensitive and cannot be delegated", and you've started testing with some AD Admin accounts.

How:

1. Check to make sure all domain controllers are running at least Windows Server 2008 R2 or later. A Windows Server 2012 R2 PDC emulator and a Windows Server 2012 R2 domain functional level is required to take advantage of all Protected Users features. Trimarc recommends Windows Server 2019 or newer for all domain controllers.
2. Make sure you've rotated passwords on your AD Admin accounts recently, especially the built-in Administrator account, so that you know everything can utilize Kerberos with AES.
3. Ensure the operational logs for Protected Users are enabled on all Domain Controllers:
    a. Open Event Viewer on Domain Controller(s)

   b. Browse to Applications and Services Logs->Microsoft->Windows->Authentication
   c. Right-click on “Protected Users: Protected User – Client” and Enable Log
   d. Right-click on “Protected User Failures – Domain Controller” and Enable Log
4. Start testing by adding one account to Protected Users and making sure there’s at least one other account to fall back to (like the break-glass built-in Administrator account) in case something goes wrong. Don’t start out by just adding the built-in Administrators group nested into Protected Users.
5. Try out all your standard AD administrative functions with the new Protected Users account. Troubleshoot any issues that arise. Do you have issues with falling back to NTLM because you’re using IP addresses or don’t have SPNs configured properly for Kerberos auth? Review the operational logs you enabled above. Check out the links below.
6. Once you have things working well for one AD Admin in Protected Users, start adding the rest of your AD Admins in phases until all are protected.
7. *Warning:* Do not add non-human service accounts or computer accounts that are nested within AD Admins to the Protected Users group. There shouldn’t be non-human accounts in AD admin groups, but we’re only tackling easy improvements in this document.

More Resources:

- https://docs.microsoft.com/en-us/windows-server/security/credentials-protection-and-management/protected-users-security-group
- https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/manage/how-to-configure-protected-accounts#BKMK_AddtoProtectedUsers

Disable Print Spooler

Disable Print Spooler service on DCs and all servers that do not perform Print services.

What: Print Spooler Nightmares

Long before PrintNightmare and PrinterBug/SpoolSample it was a great idea to limit attack surface on Domain Controllers. Part of the attack surface reduction process is disabling unnecessary services like the Print Spooler on servers that don’t print or host printers.

But what about Printer Pruning? Disabling the Print Spooler service on a DC will affect automatic printer object pruning in Active Directory. However, while Printer Pruning is enabled by default on domain controllers, it is rarely configured or even necessary. Trimarc very rarely sees Printer Pruning configured during Active Directory Security Assessments, so if you don’t see this setting in any DC-linked GPO, you’re probably not using it. If you are one of the very few organizations using the automatic printer pruning feature, losing this capability is far better than losing control of Active Directory to an attacker.

Why:

The Print Spooler is very much a legacy component in the Windows Operating System. The series of PrintNightmare vulnerabilities over the past few years demonstrated that it has a large attack surface even when patched. The no-fix PrinterBug/SpoolSample method to coerce authentication (that can be captured or relayed) from a device with Print Spooler enabled is another reason to quickly reduce attack surface on your Domain Controllers (and any other servers that don’t need to print or serve print jobs) by disabling this service. If there is Unconstrained Delegation configured in a domain, having Print Spooler running on a DC provides a very short path to full AD compromise.

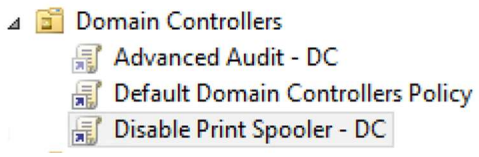
When:

Soon

How:

While the service could be disabled manually on each individual server, it's better to do so programmatically with Group Policy. Remember to use DC-specific GPOs on the Domain Controllers container. Don't mix-and-match GPOs with other OUs or use cases.

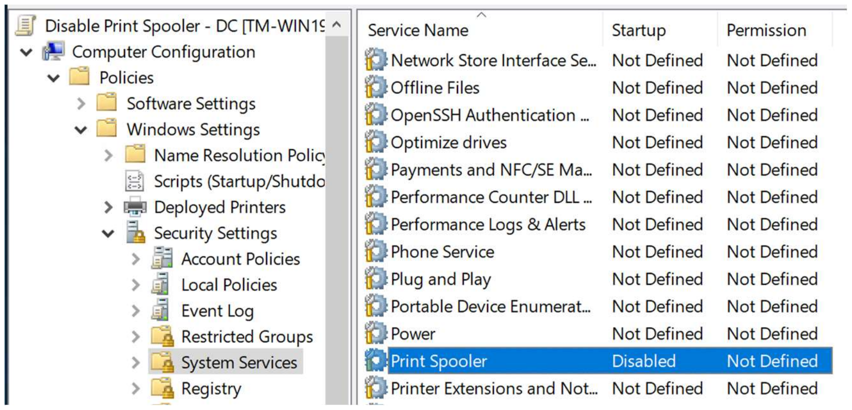
1. Link a new GPO to the Domain Controllers container:



Domain Controllers
Advanced Audit - DC
Default Domain Controllers Policy
Disable Print Spooler - DC

Figure 10 "Disaple Print Spooler – DC" GPO Applied to Domain Controllers

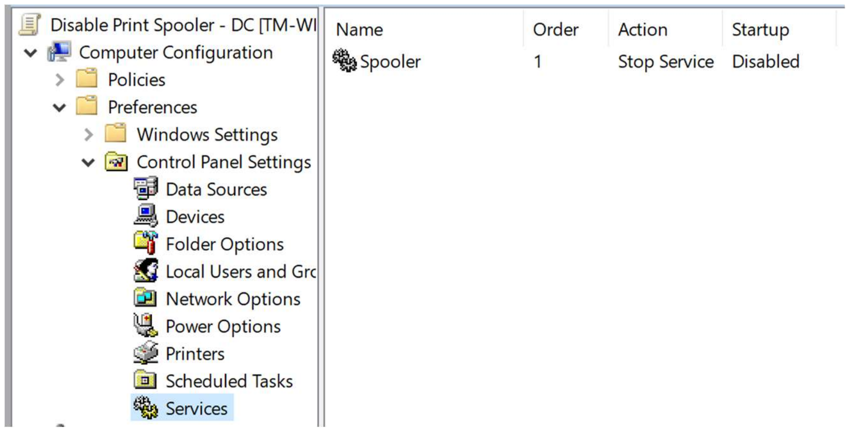
2. Utilize System Services under Policies or Services under Preferences:



Disable Print Spooler - DC [TM-WIN19
Computer Configuration
Policies
Software Settings
Windows Settings
Name Resolution Polic
Scripts (Startup/Shutdo
Deployed Printers
Security Settings
Account Policies
Local Policies
Event Log
Restricted Groups
System Services
Registry

Service Name	Startup	Permission
Network Store Interface Se...	Not Defined	Not Defined
Offline Files	Not Defined	Not Defined
OpenSSH Authentication ...	Not Defined	Not Defined
Optimize drives	Not Defined	Not Defined
Payments and NFC/SE Ma...	Not Defined	Not Defined
Performance Counter DLL ...	Not Defined	Not Defined
Performance Logs & Alerts	Not Defined	Not Defined
Phone Service	Not Defined	Not Defined
Plug and Play	Not Defined	Not Defined
Portable Device Enumerat...	Not Defined	Not Defined
Power	Not Defined	Not Defined
Print Spooler	Disabled	Not Defined
Printer Extensions and Not...	Not Defined	Not Defined

Figure 11 Disable Print Spooler Option 1



Disable Print Spooler - DC [TM-WI
Computer Configuration
Policies
Preferences
Windows Settings
Control Panel Settings
Data Sources
Devices
Folder Options
Local Users and Grc
Network Options
Power Options
Printers
Scheduled Tasks
Services

Name	Order	Action	Startup
Spooler	1	Stop Service	Disabled

Figure 12 Disable Print Spooler Option 2

3. Allow time for the new GPO to propagate and apply. Then check DCs to make sure the Spooler service is disabled.

Bonus: Check for published printer objects in AD:

```
Get-ADObject -Filter “objectCategory -eq ‘printqueue’”
```

More Resources:

- https://docs.microsoft.com/en-us/troubleshoot/windows-server/printing/use-group-policy-to-control-ad-printer
- https://adsecurity.org/?p=4056
- https://www.fortalicesolutions.com/posts/elevating-with-ntlmv1-and-the-printer-bug

Patch, Patch, Patch

Ensure DCs are running current OSes and are regularly patched.

What: Ensure DCs are running current Operating Systems

Ensure all Domain Controllers are running a currently supported version of Windows OS. Remember, Windows Server 2012 and Windows Server 2012 R2 Extended Support ends October 2023. If you aren’t running odd agents and software on your DCs, there should be few reasons for not updating to at least Windows Server 2016. Windows Server 2019 is an even better option and you could jump straight to Windows Server 2022.

Why:

Microsoft applies a Lifecycle Policy to all products to determine how long each product receives support and patches. Most on-prem operating systems used for domain controllers currently follow a Fixed Lifecycle Policy of 5 years of Mainstream Support and an additional period of Extended Support. When a product goes beyond the end of Extended Support, it will no longer receive security updates, unless the organization enrolls in and pays for the Extended Security Update Program, which is expensive and just not all that simple or easy to use.

Windows Server 2008 and Windows Server 2008 R2 left Extended Support in January 2020. Windows Server 2012 and Windows Server 2012 R2 leave Extended Support in October 2023.

When:

Check Domain Controller OS versions today using the Invoke-TrimarcADChecks.ps1 script and start deploying new DCs soon.

How:

1. Run Invoke-TrimarcADChecks.ps1 script to determine current domain controller OS versions.
    a. Ensure a domain and forest functional level of Windows Server 2008 or higher.
2. Verify that File Replication Service (FRS) is not being used for sysvol replication with dfsrmig. Windows Server 2008 and higher use Distributed File System (DFS) replication, but if an AD forest was originally created on Windows Server 2000 or Windows Server 2003 and this hasn’t been migrated, yet it could still be an issue. A status of ‘Eliminated’ indicates FRS isn’t in use.

    ```
    dfsrmig /getglobalstate
    ```

3. Add new servers running at least Windows Server 2016 (although Windows Server 2019 or Windows Server 2022 is a better choice) to the domain.
    a. Make sure the new DCs are fully patched before moving on to the next steps.
    b. Assign appropriate, meaningful computer names that follow an organization standard.
    c. Set static IP addresses.
4. Consider the complexity of your AD topology in concert with the current AD forest and domain functional levels. If the current functional levels are older than Windows Server 2008 and there are many sites or any sites with high latency it may be better to manually Adprep the forest and domains prior to promoting new domain controllers. In simple AD topologies, promoting a new DC will handle the Adprep.

```
Adprep /forestprep
Adprep /domainprep
```

5. DCPromo the new servers into DCs, wait a bit, and then perform post-DCPromo checks
    a. Check AD replication: `repadmin /showrepl /repsto`
    b. If the DC is a GC, check for GC partitions and advertisement: `Get-WinEvent -LogName "Directory Service" | Where-Object {$_.Id -eq 1119} | Format-List`
    c. Check for SYSVOL initialization & replication: `Get-WinEvent -LogName "DFS Replication" | Where-Object {$_.Id -eq 4604} | Format-List`
    d. Ensure NETLOGON & SYSVOL shares are in place: `net share`
    e. Check the Directory Service, DFS Replication, DNS Server, Application, and System Event Logs for warnings or errors and resolve.
    f. Run DCDIAG: `dcdiag /c /d /v`
6. Check Flexible Single Master Operation (FSMO) role holders and if necessary, migrate these to a newer DC in the correct site.

```
Get-ADDomain | Select-Object InfrastructureMaster, RIDMaster,
PDCEmulator
Get-ADForest | Select-Object DomainNamingMaster, SchemaMaster
```

7. Consider any clients that may be hard-coded to utilize specific DCs for LDAP authentication or DNS queries and migrate them to using new DCs. If clients are hard-coded by DC IP address, it is possible to re-use the IP address of older DCs once they have been demoted and removed from the network.
8. Demote old DCs by following Microsoft's documentation and ensure they were cleanly removed from AD DS metadata.
9. Remove demoted DCs from AD Sites and Services, if necessary.
10. Once **all** DCs are running at least Windows Server 2016, upgrade the domain and forest functional levels:

```
Set-ADDomainMode -identity <marvel.local> -DomainMode Windows2016Domain
Set-ADForestMode -Identity <marvel.local> -ForestMode Windows2016Forest
```

What: Patch DCs Consistently

Prioritize installing Microsoft security updates on Domain Controllers. Create maintenance windows for them, if needed. There have been a lot of important and critical security vulnerabilities that impact Domain Controllers in the past few years.

Domain Controllers are intended to be a single-purpose device. Any software, agent, or service considered for install on a DC should be heavily scrutinized and likely not installed at all as many will increase attack surface.

It's particularly important to not install SQL, ADFS, Azure AD Connect, SCCM Management Console, 3rd party Browsers, and outdated remote-control software on DCs.

If you must install agents on a DC, make sure they're updated and securely configured. That includes VMware Tools.

Why:

Vulnerabilities in the Domain Controller OS or any software or agents installed on DCs can be abused by attackers to gain control of AD.   Some patches fix vulnerabilities not just in the DC OS, but within Active Directory Domain Services itself. Look at CVE-2020-1472, CVE-2021-42291, the combo of CVE-2021-42287 & CVE-2021-42278 for recent examples of AD vulnerabilities that lead to AD compromise.

It's also trivial to remotely compromise a DC with versions of VMware Tools prior to version 10.1.0 installed.

When:

Monthly

How:

There are several ways to update Windows on DCs depending on the size and scope of the organization. Trimarc recommends utilizing an update method specific to DCs and other Tier 0 assets so that attack paths don't open from lower tiered systems. If your organization has only a few DCs then manual updates, WSUS, or WuFB may be appropriate. For larger organizations with hundreds of DCs it may be necessary to implement update orchestration, but that update orchestration system needs to be Tier 0-specific.

Beyond just installing Microsoft updates it's important to test updates before broadly deploying. Consider the idea of deployment rings or groups where devices are organized into 2-3 groups: Preview, Limited, and Broad.

Update/Deployment Ring	Description	Members
Preview	Gets updates early or on Patch Tuesday and reboots that night	Test Lab (hopefully)
Limited	Gets updates the weekend after Patch Tuesday. If actively exploited critical vulns, move this up depending on risk profile. Purpose is to discover potential issues without impacting operations.	1-2 DCs (small orgs) that are not FSMO role holders
Broad	Patches installed 2 weeks after Patch Tuesday. Reboot cycle staged so multiple DCs in same site do not reboot at same time. All patches installed within a month.	All remaining DCs

Sometimes organizational dynamics or policies don't allow for a "just patch" methodology. This is unfortunate, but there are paths forward. Prioritize patches by looking at the technical capability (CVSS) of the vulnerability, whether the vulnerability is known to be exploited externally (Threat Intel), and what the internal impact of the vulnerability would be on the systems (in this case DCs) if it were exploited. For example, a CVSS 9.8 being actively exploited in the wild that would lead to full AD compromise if exploited in your environment is a candidate for immediate patching.

Remember to review Microsoft update documentation as there may be registry values that need to be set to fully enable functionality of some updates.

Update VMware Tools: https://kb.vmware.com/s/article/2004754

More Resources:

- https://adsecurity.org/?p=3377
- https://docs.microsoft.com/en-us/learn/modules/active-directory-domain-services-migration/
- https://techcommunity.microsoft.com/t5/storage-at-microsoft/streamlined-migration-of-frs-to-dfsr-sysvol/ba-p/425405
- https://docs.microsoft.com/en-us/windows/deployment/update/waas-configure-wufb
- https://kb.vmware.com/s/article/2004754
- https://customerconnect.vmware.com/en/downloads/info/slug/datacenter_cloud_infrastructure/vmware_tools/12_x

Restrict Anonymous Access

Restrict Anonymous LDAP access by limiting the "Pre-Windows 2000 Compatible Access" group and confirm the dsHeuristic value is not set to 0000002.

What: Pre-Windows 2000 Compatible Access

When Active Directory Domain Services was released with Windows Server 2000, it included configurations for backwards compatibility with Windows NT and other legacy operating systems. This includes the "Pre-Windows 2000 Compatible Access" group, which Microsoft describes here:

> *The Pre-Windows 2000 Compatible Access group is used for backward compatibility for computers that are running Microsoft Windows NT 4.0 and earlier. Members of this group have Read access on all users and groups in the domain. Add users to this group only if they are running Windows NT 4.0 or earlier.*

Starting with Windows Server 2003, Microsoft started locking down anonymous Lightweight Directory Access Protocol (LDAP) connections by default but didn't change the settings on existing domains. Trimarc still finds legacy settings from domains created with Windows 2000.

Why:

When the Pre-Windows 2000 Compatible Access group contains "Everyone" or "Anonymous Logon" unauthenticated users can perform LDAP queries on Active Directory. This enables attackers to perform domain reconnaissance.

When:

Soon

How:

1. Confirm you don't have any Windows NT 4.0 servers still running in your network. ☹ l
2. Enumerate current membership of Pre-Windows 200 Compatible Access in ADUC: <domain>\Builtin, double-click Pre-Windows 2000 Compatible Access then click Members tab
3. If you are concerned about compatibility with legacy systems or Linux/Unix based operating systems, ensure that Authenticated Users is a member, although this is not necessary in most cases and does decrease AD security posture.
4. Remove ANONYMOUS LOGON and Everyone from the group.
5. Test.

What: Confirm dSHeuristics

Starting with Windows Server 2003, Microsoft configured the default dSHeuristics values to not allow anonymous LDAP as a default. The dSHeuristics attribute is used to determine the behavior of several AD components. Each character represents a different setting.

Why:

When `dSHeuristics` is configured incorrectly this provides LDAP query access to anyone on the network (even without a valid AD user account).

When:

Soon

How:

If dSHeuristics has no value (the default), anonymous LDAP is disabled.

Otherwise, anonymous LDAP is configured in the 7th character of the dSHeuristics value. A value of 2 in the 7th character enables anonymous LDAP access (bad). Any other value in the 7th character disables anonymous LDAP access (good).

Utilize AD PowerShell module to enumerate dSHeuristics:

```
(Get-ADObject -Identity ((”CN=Directory Service,CN=Windows
NT,CN=Services,CN=Configuration,” + (Get-ADDomain).DistinguishedName)) -
Properties dSHeuristics).dSHeuristics
```

The results of the snippet should either be “0” or a 7+ character value with a 7th character that is not “2”.

More Resources:

- https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-adts/7a76a403-ed8d-4c39-adb7-a3255cab82c5
- https://docs.microsoft.com/en-US/troubleshoot/windows-server/identity/anonymous-ldap-operations-active-directory-disabled
- https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-R2-and-2012/dn579255(v=ws.11)#prewindows2000-compatible-access
- https://www.stigviewer.com/stig/active_directory_domain/2013-03-12/finding/V-8547

- https://docs.microsoft.com/en-us/windows/win32/adschema/a-dsheuristics
- https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-adts/e5899be4-862e-496f-9a38-33950617d2c5

Resolve Common ADCS Misconfigurations

Review PKI objects in AD and remediate overly permissive rights.

What: Vulnerable PKI Object Access Control

In 2021, Will Schroeder and Lee Christensen of SpecterOps released Certified Pre-Owned, a collection of research into misconfigurations in Active Directory Certificate Services (ADCS). Defenders should work towards understanding the full implications of this research. However, in the meantime, there are a couple of quick wins that can help prevent the most common ADCS abuses.

Why:

When ownership and access on Certificate Authority PKI objects in Active Directory are set incorrectly it becomes possible to take over those PKI objects and potentially Active Directory itself.

When:

Soon

How:

Utilize Trimarc Senior Security Consultant Jake Hildreth's ADCS PowerShell snippets to check for these common issues:

1. Common Misconfiguration #3: Unsafe Ownership

```
$ADRoot = (Get-ADRootDSE).rootDomainNamingContext

$Safe_Owners = "Enterprise Admins|Domain Admins|Administrators"

$ADCS_Objects = Get-ADObject -Filter * -SearchBase "CN=Public Key
Services,CN=Services,CN=Configuration,$ADRoot" -SearchScope 2 -
Properties *

$ADCS_Objects | Where-Object { $_.nTSecurityDescriptor.Owner -notmatch
$Safe_Owners} | Format-Table Name,DistinguishedName
```

2. Dangerous Misconfiguration #1: Unsafe ACLs

```
#Resolve Common ADCS Misconfigurations: Unsafe ACLs - Credit Jake
Hildreth

# Review any computer IdentityReferences (trailing $) to ensure CAs are
the only computer objects to have GenericAll

# The Cert Publishers is expected, but membership of the Cert
Publishers AD group should be reviewed for unexpected membership.


$Safe_Users = "Domain Admins|Enterprise
Admins|BUILTIN\\Administrators|NT
AUTHORITY\\SYSTEM|$env:userdomain\\Administrator"


$DangerousRights = "GenericAll|WriteDacl|WriteOwner"
```

```
$ADRoot = (Get-ADRootDSE).rootDomainNamingContext

$ADCS_Objects = Get-ADObject -Filter * -SearchBase ”CN=Public Key
Services,CN=Services,CN=Configuration,$ADRoot” -SearchScope 2 -
Properties *

foreach ( $object in $ADCS_Objects ) {
    $BadACE = $object.nTSecurityDescriptor.Access | Where-Object {
        ( $_.IdentityReference -notmatch $Safe_Users ) -and (
$_.ActiveDirectoryRights -match $DangerousRights )
    }
    if ( $BadACE ) {
        Write-Host ”Object: $object” -ForegroundColor Red
        $BadACE
    }
}
```

More Resources:

- https://posts.specterops.io/certified-pre-owned-d95910965cd2
- https://github.com/TrimarcJake/adcs-snippets
- https://docs.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/dn786443(v=ws.11)
- https://github.com/GhostPack
- https://research.ifcr.dk/certipy-4-0-esc9-esc10-bloodhound-gui-new-authentication-and-request-methods-and-more-7237d88061f7

Top-Level Tier0 OU

Create Top Level OU for Admin accounts and systems. Lockdown the OU permissions and GPOs.

What: The Path to Tier 0

Tier 0 is the Control Plane for an organization’s technology. It comprises all the systems, accounts, and software used to manage the identities and access control. These are the most critical assets and should be protected as such.

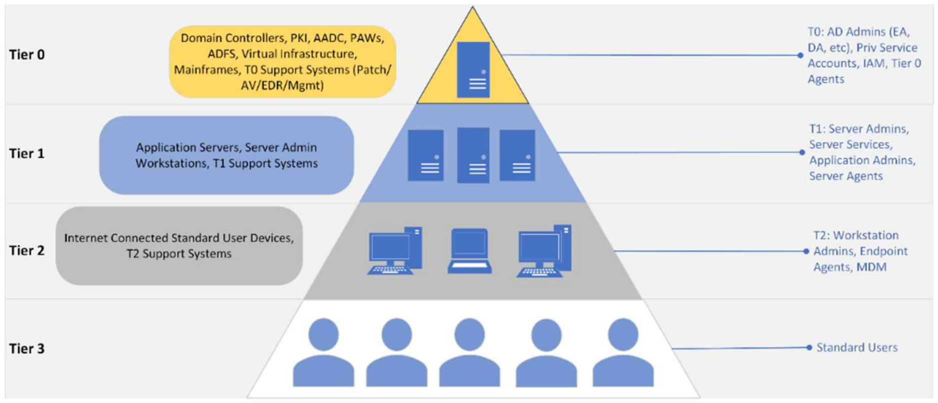

Tier 0
Domain Controllers, PKI, AADC, PAWs, ADFS, Virtual Infrastructure, Mainframes, T0 Support Systems (Patch/AV/EDR/Mgmt)
T0: AD Admins (EA, DA, etc), Priv Service Accounts, IAM, Tier 0 Agents
Tier 1
Application Servers, Server Admin Workstations, T1 Support Systems
T1: Server Admins, Server Services, Application Admins, Server Agents
Tier 2
Internet Connected Standard User Devices, T2 Support Systems
T2: Workstation Admins, Endpoint Agents, MDM
Tier 3
Standard Users

Figure 13 Trimarc Active Directory Tiering Model

If tiering does not yet exist, it's usually best to start with what's most critical and work on Tier 0. Begin by classifying every single system, security principal, and application. Or use Trimarc's classifications from the diagram above and start with the next big win: Admin OUs.

Why:

Trimarc often discovers highly privileged users, groups, and computers nested among a variety of lower privileged AD objects in OUs that have even lower privileged accounts delegated access over. Attackers discover and abuse these same findings.

Sure, `AdminSDHolder` and `SDProp` will make up for these shortcomings to a degree by not allowing inheritance on those mish-mash OUs to flow down to members of Domain Admins or Domain Controllers, but it doesn't protect the ADCS CA servers or Azure AD Connect servers that are often located in vulnerable OU structures within AD, nor the accounts with ownership and delegated control on those servers.

Tiering is intended to be a strict separation of assets and operational procedures to administer them whereby a lower tier account cannot be used to manage higher tier devices. In maintaining that separation, it's also important to not manage down, such as using members of Domain Admins to log into user workstations, as this exposes higher tier identities to a lower tier.

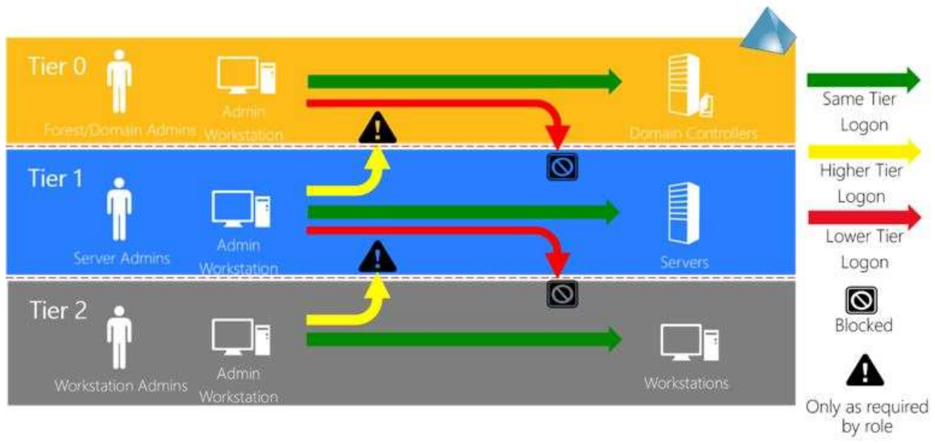


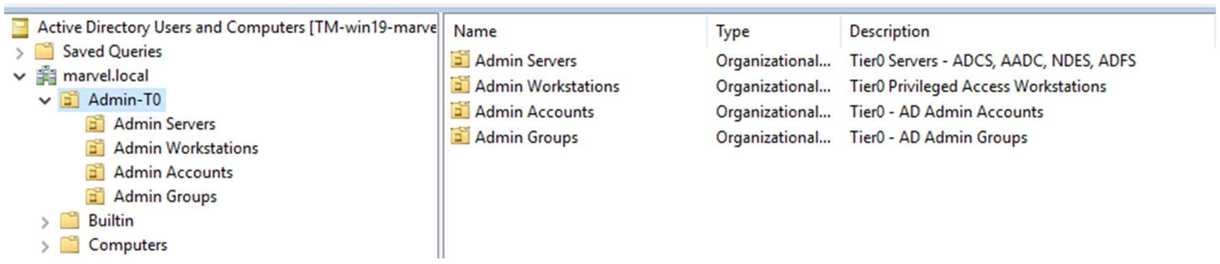
Figure 14 Active Directory tiered administrative model logon restrictions (Image Credit: Microsoft)

When:

Soon. Review at least annually to make sure no Tier 0 devices or accounts snuck out of the Admin OU.

How:

1. Take a good look at your environment and figure out everything that is Tier 0. Create an inventory. You can't protect what you don't know you have.
2. Create a top-level OU called something like "Admin" or "Tier0", which only AD Admins can access and control.



Name	Type	Description
Admin Servers	Organizational...	Tier0 Servers - ADCS, AADC, NDES, ADFS
Admin Workstations	Organizational...	Tier0 Privileged Access Workstations
Admin Accounts	Organizational...	Tier0 - AD Admin Accounts
Admin Groups	Organizational...	Tier0 - AD Admin Groups

Figure 15 Root-level Admin OU

   a. Modify OU Security so Authenticated Users don't have view access.
   b. Block GPO Inheritance. Create, apply, & link Admin OU specific GPOs.
   c. Create child OUs
      i. Admin Servers
      ii. Admin Workstations
      iii. Admin Accounts
      iv. Admin Groups
   d. **Only** AD Admins have Modify Rights to the Admin OU structure.
   e. **Only** AD Admins have Modify/Owner rights to GPOs linked to this OU structure.
3. Place all AD Admin related objects (users/groups) in this Admin OU structure.
   a. Note: Default groups are expected to be in their default location, use caution when moving them
4. Leverage AD Admin Servers (good) or Privileged Access Workstations (better) for any admin tasks.
   a. AD Admin Servers are locked down RDP hosts that requires MFA for AD Admins to connect
   b. Privileged Access Workstations (PAW) are locked down PCs (or secured VM) that are only used for Tier0 admin tasks
5. Place all AD Admin systems (any Tier 0 computer accounts that aren't DCs) in the Admin OU structure.
6. If you start out with an AD Admin Server, keep pushing forward to implement PAWs.

More Resources:

https://docs.microsoft.com/en-us/security/compass/overview

Conclusions

The reality here is that nothing about properly securing an on-prem Active Directory environment is quick or easy. However, like any journey, it begins with that first step. Throughout this paper we've given you lots of options for that step. Some steps are tiny, and some are giant leaps. The step you choose will depend on your appetite and ability for change.

If you can't quite figure out what size step you need, give Trimarc's ADChecks PowerShell Script a try. It will capture some of the same information that Trimarc uses to perform our Active Directory Security Assessment (ADSA) engagements.

Additional Resources

- Trimarc Content Hub: https://www.hub.trimarcsecurity.com/
- Trimarc ADChecks PowerShell Script: https://www.hub.trimarcsecurity.com/post/securing-active-directory-performing-an-active-directory-security-review
- Trimarc Active Directory Security Assessment: https://www.trimarcsecurity.com/ad-security-assessment
- Sean Metcalf's Personal Blog: https://adsecurity.org/
- Microsoft's Best Practices for Securing Active Directory: https://docs.microsoft.com/en-us/windows-server/identity/ad-ds/plan/security-best-practices/best-practices-for-securing-active-directory
- Microsoft's Detecting & Preventing Privilege Escalation Attacks Leveraging Kerberos Relaying: https://www.microsoft.com/security/blog/2022/05/25/detecting-and-preventing-privilege-escalation-attacks-leveraging-kerberos-relaying-krbrelayup/
- Microsoft Security Advisory 974926: https://docs.microsoft.com/en-us/security-updates/SecurityAdvisories/2009/974926
- Monitoring What Matters – WEF For Everyone: https://docs.microsoft.com/en-us/archive/blogs/jepayne/monitoring-what-matters-windows-event-forwarding-for-everyone-even-if-you-already-have-a-siem

TRIMARC